Computing Just What You Need: Online Data Analysis and Reduction at Extreme Scales

Ian Foster[1,2(✉)], Mark Ainsworth[3], Bryce Allen[2], Julie Bessac[1], Franck Cappello[1], Jong Youl Choi[4], Emil Constantinescu[1], Philip E. Davis[5], Sheng Di[1], Wendy Di[1], Hanqi Guo[1], Scott Klasky[3], Kerstin Kleese Van Dam[6], Tahsin Kurc[7], Qing Liu[8], Abid Malik[6], Kshitij Mehta[4], Klaus Mueller[7], Todd Munson[1,2], George Ostouchov[4], Manish Parashar[5], Tom Peterka[1], Line Pouchard[6], Dingwen Tao[1], Ozan Tugluk[3], Stefan Wild[1], Matthew Wolf[3], Justin M. Wozniak[1], Wei Xu[6], and Shinjae Yoo[6]

[1] Argonne National Laboratory, Lemont, IL, USA
foster@anl.gov
[2] University of Chicago, Chicago, IL, USA
[3] Brown University, Providence, RI, USA
[4] Oak Ridge National Laboratory, Oak Ridge, TN, USA
[5] Rutgers University, New Brunswick, NJ, USA
[6] Brookhaven National Laboratory, Brookhaven, NY, USA
[7] Stony Brook University, Stony Brook, NY, USA
[8] New Jersey Institute of Technology, Newark, NJ, USA

Abstract. A growing disparity between supercomputer computation speeds and I/O rates makes it increasingly infeasible for applications to save all results for offline analysis. Instead, applications must analyze and reduce data online so as to output only those results needed to answer target scientific question(s). This change in focus complicates application and experiment design and introduces algorithmic, implementation, and programming model challenges that are unfamiliar to many scientists and that have major implications for the design of various elements of supercomputer systems. We review these challenges and describe methods and tools that we are developing to enable experimental exploration of algorithmic, software, and system design alternatives.

1 Introduction

Technology trends are creating a crisis in high performance computing. Computer speeds are increasing much faster than are storage technology capacities and I/O rates. For example, the Mira supercomputer installed at Argonne National Laboratory in 2012 has a peak compute rate of 10 petaflop/s (10^{16} op/s) and disk write rate of 500 GB/s (5×10^{11} bytes/s). By 2024, computers are projected to compute at 10^{18} ops/sec but write to disk only at 10^{12} bytes/sec: a compute-to-output ratio 50 times worse. Figure 1 provides another perspective on this trend. We can no longer output every piece of information that we might ever possibly *want*. Instead, we need to output just the information that

F.F. Rivera et al. (Eds.): Euro-Par 2017, LNCS 10417, pp. 3–19, 2017.
DOI: 10.1007/978-3-319-64203-1_1

we *need* to answer some question(s). This new goal requires new thinking about the design and implementation of both applications and system software.

In both purely computational and coupled experimental–computational studies, these growing disparities between computational speeds and I/O rates demand new application structures that link previously disjoint activities: experiment, simulation, data analysis, data reduction. Yet while many algorithms and tools exist to treat separate pieces of such problems, these capabilities are often inoperable or inaccessible to the research scientist. Scientists need new tools for coupling components and new methods for co-optimizing the resulting workflows. These tasks introduce algorithmic, implementation, and programming model challenges that are unfamiliar to many scientists and that have major implications for the design of various elements of high performance systems.

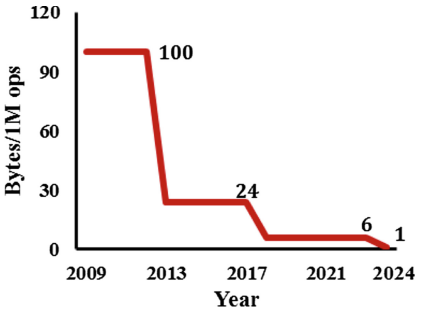

Fig. 1. Total filesystem throughput of leadership class facilities vs. total floating point operations per second [25,32,40]. I/O throughput scales more slowly than computational speed.

The emerging exascale landscape offers many opportunities to address these problems. Additional storage features such as non-volatile random access memory (NVRAM) will provide powerful caching and aggregation capabilities. A variety of operating systems, runtime, scheduling, and fault tolerance features may become available to applications and middleware developers. Advanced workflow systems, I/O frameworks, and data reduction techniques can be integrated to construct efficient data processing pipelines. These features will be adopted by a range of exascale-ready applications, so there is a unique window of opportunity to develop solutions that are widely applicable, reusable, and beneficial.

The Co-design center for Online Data Analysis and Reduction (CODAR) engages scientists at three national laboratories and five partner universities, to address these challenges. Working closely with applications teams, CODAR is undertaking a co-design process that targets both common data analysis and reduction methods (e.g., feature and outlier detection, and compression) and methods specific to particular data types and domains (e.g., particle and structured finite-element methods). Our goal is to understand and guide tradeoffs in the development of computer systems, applications, and software frameworks, given constraints relating to application development costs and fidelity, performance portability, scalability, and power efficiency, and to answer these questions:

Q1: What are the best data analysis and reduction algorithms for different application classes, in terms of speed, accuracy, and resource needs? How can we implement those algorithms for scalability and performance portability?

Q2: What are the tradeoffs in data analysis accuracy, resource needs, and overall performance between online reduction and offline analysis vs. online analysis? How do these tradeoffs vary with hardware and software choices?
Q3: How do we effectively orchestrate online data analysis and reduction to reduce associated overheads? How can hardware and software help?

2 Related Work and Context

We are not the first to observe that both the growing disparity between compute and I/O rates and the need for near-real-time feedback requires online data analysis and reduction. Much work has been performed on "in-situ" and "in-transit" analysis methods [4,9], motivated by a desire to conserve I/O bandwidth, storage, and/or power; increase accuracy of data analysis results; and/or make optimal use of parallel platforms [31], among other factors [2]. The need to reduce output data volumes has also spurred various science teams to create custom online data analysis and reduction techniques [16,22,26,27,34,36] and also stimulated work on general-purpose methods [8,10,21].

Such work reveals complex relationships between application design, data analysis and reduction methods, programming models, system software, hardware, and other elements of extreme-scale systems, particularly given constraints such as applicability, fidelity, performance portability, and power efficiency.

The community is far from completely understanding the many co-design issues posed by online data analysis and reduction. For the broader community to leverage and expand the knowledge gained by early adopters, they will require an effective, usable and sustainable software infrastructure that allows scientists to use the *best techniques* to extract the *right information* that can then be pushed through the straw to the parallel file system. It is in this context that we established the CODAR co-design project.

3 Example Applications

We use examples from climate, fusion, and materials science to motivate the need for online data analysis and reduction.

3.1 Climate Science

Climate scientists want to run large ensembles of high-fidelity 1 km × 1 km simulations on exascale systems, with each instance simulating 15 years of climate in 24 h of computing time. They estimate that outputting the full model state for each ensemble member once per simulated day would generate 260 TB every 16 s across the ensemble, approximately 16× what can be written to the parallel file system at the expected peak output rate of 1 TB/sec. (Currently, climate models achieve much lower I/O rates, due to their relatively small model grids.) Furthermore, even following data reduction to 1 TB/sec, such runs would output 85 PB per day, posing major storage and offline data analysis challenges.

While 85 PB is a lot of data to output in a day's computing, this quantity represents just a small subset of the total data to be produced by the ensemble. Outputting state just once per simulated day represents a highly lossy reduction, given that the climate model time step may be just 100 simulated seconds, and indeed some analyses may require access to the full state at higher frequency. For example, feature detection (e.g., tracking cyclones, detecting areas of extreme heat) may require access to model state once per simulated five minutes, a rate $24 \times 12 = 288$ times greater. Clearly, climate models need new online data analysis and reduction methods that can both preserve more information than once-per-day snapshots and produce considerably less data.

3.2 Fusion Science

Fusion scientists are developing a high-fidelity whole device model for magnetically confined fusion plasmas, for use in planning experiments on the ITER facility and simulating future experimental fusion devices [6]. The X-point included Gyrokinetics Code (XGC) [24], one potential component of a whole device model, models the plasma edge. A single XGC simulation can produce hundreds of petabytes of data describing particle positions and the state of the field within which the particles move.

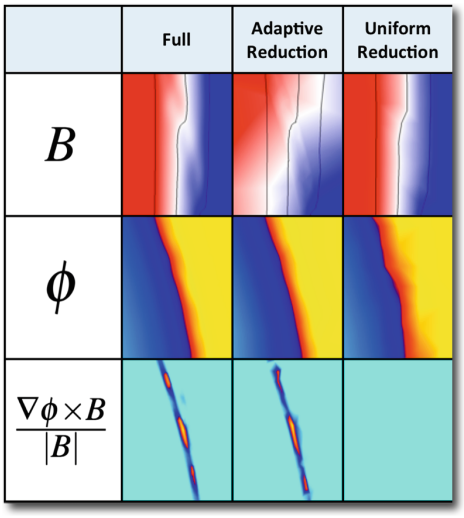

Fig. 2. XGC fusion simulation results near the plasma edge illustrates the need for fidelity preserving data reduction. The full data for the magnetic field $||B||$ and the scalar potential ϕ both show close approximation to the full solution. However, in the case of the derived fluid velocity $\frac{\nabla\phi\times B}{||B||}$, the adaptive method retains the four major features from the full data; the uniform method does not.

We use this example to illustrate the need for application-aware data reduction methods. To reduce this data to manageable sizes, ultimately allowing 100 PB to be reduced to 100 TB, a 1000:1 reduction, fusion scientists and CODAR participants collaborated to devise a multistep data reduction process. The first step was to simply decrease output frequency. However, this approach cannot be taken beyond physically relevant time scales; important information would be lost by decreasing the frequency further. The second step was to use application knowledge to further reduce the data without losing essential information. The XGC particles are assumed to follow a Maxwellian distribution. Therefore, we fitted a distribution to the data and saved the parameters for the distribution and the particles falling outside that distribution (the "outliers"). For the field data, adaptive data reduction methods

were used to preserve features (see Fig. 2). Finally, generic compression methods were applied to achieve further data reduction. The reduced data was then output and used for offline data analysis.

3.3 Materials Science

Materials scientists regularly run billion-atom atomistic simulations with femtosecond time steps on leadership-class machines [33,37]. In order to understand phenomena such as the structural properties of lignin-based macromolecules, information essential for improving biofuel production, measurable vibrational responses that arise at the tens of femtoseconds must be studied, requiring per-time step data access. Yet folding and bonding properties arise only on the scale of seconds. Saving the full state to simultaneously study both quantities would generate exabytes on exascale computers. Intelligent, statistically valid spatial and temporal data analyses and/or reductions that can be applied online are needed to achieve accurate scientific characterizations with reduced data.

3.4 Real-Time Decisions and Data Assimilation

Increasing use of supercomputers for near-real-time decision making is another factor motivating new thinking about application and system software design [1]. For example, both experimental fusion energy experiments and next-generation light sources are moving to a new frontier where data must be processed rapidly to enable near-real-time decisions.

In light source science, high-fidelity simulation models are used to fit parameters that describe sample structure [11]. Coupling emerging high-frame-rate, high-resolution detectors with high-performance computing and networks allows these models to be calibrated by streaming data from the experiment hall. Future experiments may also be guided by active learning methods that prioritize observations that reduce error and uncertainty in the model. Due to the growth in detector and simulation capabilities, it is no longer feasible to input experimental data, perform some computation (e.g., simulation of the experiment's future trajectory), and store results for later analysis. Data must be transmitted and assimilated immediately to maximize the quality of the simulated model, process significant events, and/or permit rapid feedback to the experiment.

International experimental fusion energy experiments are moving to a new frontier where data needs to be processed as soon as possible to make near-real-time decisions. Data sizes, rates, and durations are increasing faster than Moore's Law, and new software technologies are needed to cope with their ability to do their science quickly and accurately. One critical challenges is to understand which data need to be processed immediately (in near real rime), and we need the ability to express this during the data generation, and to compose a workflow that can help scientist get the best out of their data with a given amount of work.

4 A High-Performance Co-design Architecture

Some science teams have already developed application-specific online data analysis and/or reduction methods on petascale systems, methods they now need to scale for exascale. Others face the prospect of having to integrate such methods from scratch as part of their preparation for exascale. In both cases, we want to make it easy for them to integrate a variety of scalable online data analysis and reduction methods into their existing infrastructure, so that they can easily experiment with co-design alternatives and achieve performance portability.

4.1 The Need for Modular Implementations

A first key to achieving this goal of easy co-design, we argue, is to modularize implementations so that analysis and reduction methods, resource allocations, and coupling methods can be varied with little or no changes to an application. In this way we facilitate experimentation with design alternatives and investigation of co-design and performance portability questions.

The key to modular integration of applications with online data analysis and reduction methods is access to both the application data of interest and metadata describing that data's structure. Once this access is enabled, it becomes straightforward to access and exchange the data to be analyzed and/or reduced. Our team has much experience in instrumenting applications to provide and use such information, particularly in the context of the Adaptable IO System (ADIOS) [17,43], the Swift [3,42] system, and in earlier work [12,44]. In many cases, this instrumentation involves adding simple procedure calls, for example via the ADIOS application program interface (API) [28], to the application to indicate the data structures in question. A runtime system can then extract the specified data and pass it to specified data analysis and reduction services. Only the runtime implementation, not the application, needs to be modified to explore alternative implementation strategies, such as processing on the same or different nodes, using NVRAM, varying clock speeds for power efficiency, or varying the number of data analysis nodes.

The characteristics of data in extreme scale simulations will be highly dynamic with respect to volume and relative importance. For example, the detection of a rare event in a simulation could trigger analysis of additional complexity, or even require a different analysis routine to be loaded and executed. Thus, the runtime must be highly reconfigurable. It must enable the user to programmatically branch into new analysis pipelines or rebalance resources among components. This will require a novel integration of high-level directives and hints with low-level I/O reconfiguration features to allow the overall workflow to adapt to conditions that emerge during execution.

4.2 CODAR System Components

These considerations lead us to identify three major classes of CODAR co-designed technologies. Figure 3 shows how they fit together.

First, the **CODAR Data API** allows applications to specify the data to be analyzed and/or reduced, and its structure. We leverage the ADIOS API [23], which has been integrated into more than thirty science applications [17,30,38,43]. One co-design question will be how to extend this API so that applications can convey actionable information for exascale optimizations relating to performance and power efficiency.

Second, **CODAR Data Services** provide scalable implementations of data analysis and reduction methods, plus ancillary monitoring methods, all packaged to permit their use by any application. The data reduction methods will provide effective reduction of the simulation outputs, both the application state and the results of online data analyses applied to that state, while retaining simulation fidelity. Data monitoring is needed to verify that a particular data reduction method is retaining the necessary information and to provide feedback when the data reduction is either too aggressive or not aggressive enough (see Fig. 2). These services will include a mix of those developed by us and those imported from other sources. We are developing only a modest number of such implementations ourselves, but our methods and co-design knowledge will be broadly applicable. Anyone will be able to add generic or application-specific data services. An important co-design question here concerns the methods and support required for efficient execution of a broad range of such services.

Third, the **CODAR Runtime** provides methods for the deployment, configuration, execution, and computational monitoring of applications and associated data analysis and reduction pipelines on exascale platforms. Given a specified set of data analysis, reduction, and monitoring services, it will enable their efficient composition and configuration; their deployment to appropriate nodes and cores; efficient communication among them; computational monitoring of both individual services and the complete computation; and adaptation of service configurations and parameters.

We intend that these three co-designed technologies allow application teams to instantiate versions of the Fig. 3 pipeline to address their specific science

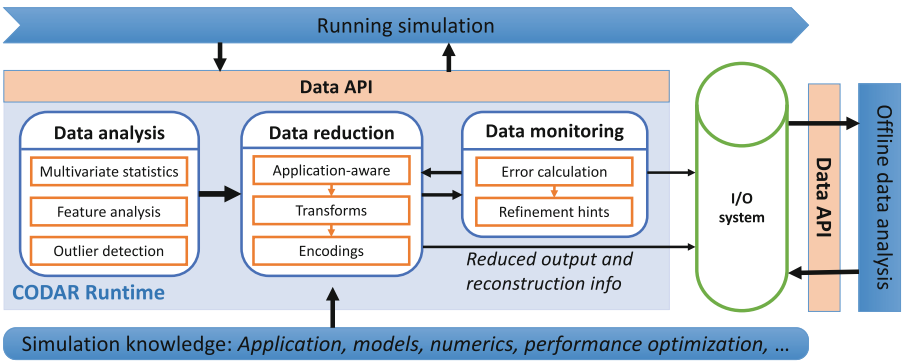

Fig. 3. Prototypical data analysis and reduction pipeline, showing how a simulation communicates to our services through an API that conveys data and their structure.

goals. Lessons learned from experiments with diverse applications, methods, and platforms will in turn feed back to ECP application projects, software projects, vendors, and other stakeholders.

5 CODAR Data Services

We show in Fig. 3 a prototypical application pipeline in which an online data analysis service consumes simulation data and produces extracted information that is communicated back to the simulation and/or sent to a data reduction service for further processing prior to storage on a parallel file system. A monitoring service can also be engaged to evaluate the quality of the data reduction results. More generally, data analysis methods may extract information from several states—for example, a sliding window of time—and use results from previous data analyses. We review some of the analysis, reduction, and monitoring methods that we are studying in the CODAR project.

5.1 Analysis Services

Our initial catalog of data analysis methods concentrates on multidimensional statistical and image analysis and outlier detection and extraction. We develop this set based on application requirements and their relevance to important co-design questions, such as the following. When should a data analysis be performed online versus offline? How frequently can data analyses be performed online, given a specified computational budget? How can data analyses make use of increased CPU on-node concurrency? When do we use burst buffers to stage and extend memory for online data analysis? How do we take advantage of deep memory hierarchies for tracking changes over time?

Multidimensional Statistical Analysis. Application scientists frequently find it useful to extract multidimensional statistics and geometrical characteristics from simulations, since these analyses reflect properties on a larger scale than do pointwise and time-instant measurements, and carry information about structures, aggregated quantities, and statistical measurements.

We plan to build on our stochastic flow map [15], which provides understanding of uncertain transport behavior. This map has been successfully applied to climate [15,35] and weather [14] applications. We are further developing our data analysis methods to model multivariate and multiscale features in statistical ensembles using the concepts of specific mutual information between variables [5] and information flows based on association rules [29]. These methods all have a wide range of applications including climate and combustion.

As an example, climate model ensembles produce a distribution of velocities, instead of a single velocity at each grid point. These distributions allow climate scientists to quantify the uncertainty in convergent and divergent transport behaviors and in derived features such as eddies, flow segmentation, and large-scale teleconnections. Tracking these features via stochastic flow maps enables scientists to understand their evolution and advance their scientific mission.

Outlier Detection and Extraction. Outliers and rare events are the needles that application scientists frequently seek in the massive haystack of exascale data. We are developing semi-supervised machine learning techniques that incorporate existing prior knowledge (such as a Bayes classifier) within an unsupervised learning algorithm to select the most relevant targets for later inspection and addition to a corpus of information. We are integrating the iForest [20] unsupervised machine learning algorithm to project data into a subspace where outliers deviate sharply from the remaining data, and applying kernel-based signatures to detect outliers [18–20]. This combination is particularly effective in the case of complex data with extremely high dimensionality [19,20].

5.2 Reduction Services

As illustrated in Sect. 3, the communication, analysis and storage of data from exascale simulations will only be possible through aggressive data reduction capable of shrinking datasets by one or more orders of magnitude. Such data reduction level is not feasible with lossless data reduction (e.g., lossless compression) that only typically achieve reduction factors of 2 (initial size/reduced size) on scientific data. Only lossy data reduction has the potential to reach reduction factors of orders of magnitude.

As shown in Fig. 3, online data reduction services consume both simulation outputs and the results from online data analyses and prepare data to be written to a file system. A crude but commonly used data reduction technique is to save data only periodically (e.g., every n-th time step) and use linear interpolation to approximate the missing values for offline data analysis. This technique can achieve arbitrary reduction ratios, but it lacks control over the errors. While we support this technique, our data reduction goal is to preserve the essential information in the reduced output while satisfying resource constraints on I/O bandwidth. Thus, we need reduction methods that provide control over errors.

The consumer big data domain is in advance of science in the systematic use of lossy data reduction. Most photos taken on a smartphone are stored in lossy compressed form, as are audio and video files. The projection made by CISCO about the Internet traffic is striking: in 2025, 80% of the Internet traffic will be video streaming; which means that more than 80% of the data transiting on the Internet will be lossy compressed. Microsoft has already deployed FPGAs into its data centers to accelerate JPEG compression (among other operations). An important distinction between the scientific and consumer big data domains is the specificity of the data reduction techniques. The consumer big data domain relies on generic lossy compressors (e.g., JPEG for images, MP3 for audio and MPEG4 for video). Many scientific applications at extreme scale already need aggressive data reduction. Spatial sampling and decimation in time are used to reduce data but these techniques also reduce significantly the quality of the data analytics performed on the sampled or decimated datasets. Advanced lossy compression techniques provide a solution to this problem by allowing the user to better control the data reduction error. However, the adoption of lossy data

reduction techniques in the scientific domain is still limited because of the lack of comprehensive understanding of the errors introduced by lossy data reduction.

Although lossy data reduction is critical to evolve many scientific domains to the next step, the technology of scientific data reduction and the understanding on how to use it are still in their infancy. The first evidence is the lack of results in this domain: over the 26 years of the prestigious IEEE Data Compression Conferences, only 12 papers identify an aspect of scientific data in their title (floating-point data, data from simulation, numerical data, scientific data). The second evidence is the poor data lossy reduction performance on some datasets. Beyond the research on data reduction techniques, scientists also need to understand how to use lossy data reduction. The classic features of compressors (integer data compression, floating-point data compression, fast compression and decompression, error bounds for lossy compressors) do not characterize data reduction algorithms specifically with respect to their integration into a high-performance computing and data analytics workflow.

The CODAR co-design project is addressing these two gaps by collecting data reduction need from exascale application, investigating and developing new lossy data reduction algorithms, collecting error assessment needs from applications and developing a tool, called Z-checker, to assess comprehensively the error introduced by lossy data reduction.

One approach to lossy scientific data reduction is for application and system developers to design application-specific lossy data reduction technique. This approach is used, for example, at the Large Hadron Collider, where experiments use specialized hardware and software to extract only "interesting" events from TB/s data streams. An alternative approach is to design and use generic lossy compressors for scientific data. Several teams have worked and are still working on this problem. The difficulty here is to develop lossy compressors that provide excellent data reduction performance for a large variety of scientific applications: regular mesh, irregular mesh, particle simulation, instrument, etc.

Appropriately chosen reduction methods can improve the information content of output data. For example, the FLASH hydrodynamics simulation code [13] is widely used to perform extremely large simulations. Conventionally, data are not output every time step, the remaining data are discarded. An alternative curve-fitting technique exploits the fact that hydrodynamic flows are mostly smooth and thus can be greatly reduced by lossy compressors that nevertheless provide error bounds. Our SZ compressor [8], for example, can achieve 100:1 reduction for the BLAST2 hydrodynamics data [7].

Currently, the two leading lossy compressors for scientific data are SZ [8,39] and ZFP [41]. They are error-bounded lossy compressors, meaning that they respect user-specified error constraints. Each uses a completely different compression strategy. One is based on a prediction method and the other one is transform based. One is better than the other, depending on the application and the dataset. Research in this domain aims to reach compression factors of 10 for hard to compress datasets and >100 for easy to compress ones. These two lossy compressors as well as other generic lossy compressors for scientific

data work well for smooth datasets. They are less effective when the datasets are very irregular and presents large variations. One important aspect of the CODAR project is to understand what compression algorithm (or sequence of algorithms) to use according to the characteristics of the datasets. We return to this question in the next section.

5.3 Monitoring Services

Scientific and consumer big data are distinguished by their quite different quality requirements for reduced datasets. JPEG, MP3 and MPEG4 are not only generic but universal: all users have the same perception of images and sound. Thus, compression quality criteria can be defined that meet the needs of a large population of users. In science, on the other hand, each combination of application and data may involve different quality requirements. One open question is the relevant set of quality criteria for scientific datasets. As illustrated in Fig. 2, blindly applying a reduction method can result in a failure to capture features that are essential for subsequent analysis. Users have already expressed needs to assess spectral alteration, correlation alteration, the statistical properties of the compression error, the alteration of first and second order derivatives, and more. As the domain of lossy data reduction for scientific datasets grows, the community will learn what metrics are relevant and needed.

Another open question is how to express quality requirements, in particular when there are many such requirements with interdependencies. Perhaps the most important open question is the comprehensive assessment of the error introduced by lossy data reduction. The classic lossy compressor assessment metrics, PSNR (peak signal to noise ratio) and its extension, the rate distortion diagram, are not enough to represent the potential impact of the error on scientific datasets and the analyses that may be performed on them. Users may also be interested in other distortions (spectral, derivative, distribution) and other characterization of the error (autocorrelation, distribution).

To address these concerns, we are developing data monitoring services for estimating data reduction errors and providing (1) feedback to the reduction methods so that their tolerances can be adjusted and (2) reduction error maps for the application scientist. These maps can be imported into offline data analysis routines or visualized to observe the evolution of reduction errors.

Our first step is a simple monitoring service, Z-checker, that applies an extensible set of metrics to assess both initial dataset properties and the alterations introduced by lossy data reduction. The Z-checker is designed to permit the integration of a wide range of analysis modules, in C, C++, Fortran, and R. An initial set allow its use to characterize critical properties (such as entropy, distribution, power spectrum, principle component analysis, auto-correlation) of any dataset to improve compression strategies, detect the compression quality (compression ratio, bit-rate), and provide global distortion analysis comparing the original data with the decompressed data (peak signal-to-noise ratio, normalized mean square error, rate-distortion, rate-compression error, spectral,

distribution, derivatives) and statistical analysis of the compression error (maximum/minimum/average error, autocorrelation, distribution of errors). Our initial Z-checker runs offline; it will evolve into an online application that can be configured to run multiple user-specified analyses concurrently, either for the purpose of online steering of data reduction or to produce assessment reports that can be used to evaluate reduction performance under different settings.

As we gain experience with online use of Z-checker, important questions to be answered include the following. How frequently should we estimate the reduction error? What data analysis methods and metrics should we use for this estimation? How quickly can we provide the refinement hints so that the information provided is actionable? How effective are the refinement hints at influencing the reduction error?

6 The CODAR Runtime

The CODAR Runtime provides methods for controlling the placement and configuration of CODAR Data Services for purposes of co-design exploration and performance optimization. The initial focus is on simple manual configuration of service delivery choices; in later stages of the project, we will also provide for automated configuration, once co-design strategies are better understood. Figure 4 shows the initial set of components.

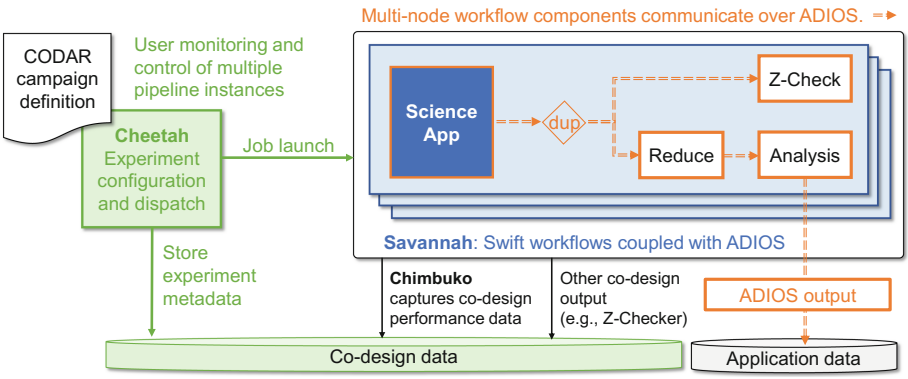

Fig. 4. The CODAR co-design system, showing in particular the Cheetah experiment management component and the Savannah runtime.

The **Cheetah** experiment management framework defines a set of conventions and re-usable scripts for conducting parameter sweep experiments on different science applications. Such experiments are intended to be run on supercomputers, particularly on existing machines, but may also be run on local workstations for debugging. An 'application' may be a single science code or, more typically, one or more science codes plus a set of online analysis and reduction

codes that are coupled with the science codes and each other. The goal of such parameter sweep experiments is to determine the best set of parameters to use to run the application as efficiently as possible on different target machines. This 'best' set of parameters usually varies over different machines.

The **Savannah** in situ runtime serves three purposes. It provides a tested deployment framework for any application (or software technology) project to use online data analysis and reduction; provides the infrastructure needed to create a testing framework (Cheetah) to evaluate reduction and analysis functions for performance on a variety of levels (application and platform); and provides a reference approach for teams that have specialized needs that exceed the infrastructure design constraints

Savannah is not intended to be the only possible way of deploying CODAR-developed or vetted analytics and reduction functions; multiple cooperating ecosystems are needed to make the total system thrive. However, Savannah offers a convenient and straight-forward approach, making it easier for applications to focus on the science, rather than the details of advanced scheduler settings, RDMA network transfers, and other technical details that tend to interfere with the deployment of online techniques.

Finally, the **Chimbuko** performance data capture suite captures, analyzes and visualizes performance metrics for complex scientific workflows and relates these metrics to the context of their execution on extreme-scale machines to enable empirical performance studies. Because capturing performance metrics can quickly escalate in volume and provenance can be highly verbose, Chimbuko interfaces with (lossy) data compression modules specialized for high-velocity performance data.

To quantify co-design tradeoffs involved in online data analysis and reduction for a particular application, an ensemble of executions would be run using Cheetah and Savannah, each involving an application X plus an analysis A and a reduction R (e.g., from Z-checker) with different specifications of the information that needs to be saved when (e.g., different data reduction mechanisms and parameters) and what work is to be placed where (e.g., different numbers of nodes allocated to X, A, and R; X, A, and R allocated to the same or different nodes; and different mechanisms used to transfer data between components). Chimbuko would capture the performance information for each member of the ensemble and enable analysis across the ensemble to answer co-design questions.

7 Conclusion

We have presented the rationale, technical approach, and some initial results for the new Co-design center for Online Data Analysis and Reduction (CODAR). This project is motivated by the growing disparity between compute and I/O speeds on high-performance computers, and the consequent need to perform data analyses and reductions increasingly online, while an application is running, rather than offline. Such new computational structures in turn lead to new co-design questions, such as which analysis and reduction methods to use in different

contexts, how to construct such application–analysis–reduction computations, and how to map and configure different components. CODAR is developing new methods that will allow the principled investigation of such questions.

Acknowledgments. This research was supported in part by the Exascale Computing Project (17-SC-20-SC) of the U.S. Department of Energy (DOE), and by DOE's Advanced Scientific Research Office (ASCR) under contract DE-AC02-06CH11357.

References

1. Future Online Analysis Platform Workshop, April 2017. https://press3.mcs.anl.gov/futureplatform/about
2. Ahrens, J.: Increasing scientific data insights about exascale class simulations under power and storage constraints. IEEE Comput. Graph. Appl. **35**(2), 8–11 (2015)
3. Armstrong, T.G., Wozniak, J.M., Wilde, M., Foster, I.T.: Compiler techniques for massively scalable implicit task parallelism. In: International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2014 (2014)
4. Bauer, A.C., Abbasi, H., Ahrens, J., et al.: In situ methods, infrastructures, and applications on high performance computing platforms. Comput. Graph. Forum **35**(3), 577–597 (2016)
5. Biswas, A., Dutta, S., Shen, H.W., Woodring, J.: An information-aware framework for exploring multivariate data sets. IEEE Trans. Vis. Comput. Graph. **19**(12), 2683–2692 (2013)
6. Bonoli, P., McInnes, L.C.: Report of the workshop on integrated simulations for magnetic fusion energy sciences (2015). https://www.burningplasma.org/resources/ref/Workshops2015/IS/ISFusionWorkshopReport.11.12.2015.pdf
7. Colella, P., Woodward, P.R.: The piecewise parabolic method (PPM) for gas-dynamical simulations. J. Comput. Phys. **54**(1), 174–201 (1984)
8. Di, S., Cappello, F.: Fast error-bounded lossy HPC data compression with SZ. In: IEEE International Parallel and Distributed Processing Symposium, pp. 730–739 (2016)
9. Dorier, M., Dreher, M., Peterka, T., Wozniak, J.M., Antoniu, G., Raffin, B.: Lessons learned from building in situ coupling frameworks. In: 1st Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization, pp. 19–24. ACM (2015)
10. Dreher, M., Raffin, B.: A flexible framework for asynchronous in situ and in transit analytics for scientific simulations. In: 14th International Symposium on Cluster, Cloud and Grid Computing, pp. 277–286. IEEE (2014)
11. Foster, I., Ananthakrishnan, R., Blaiszik, B., Chard, K., Osborn, R., Tuecke, S., Wilde, M., Wozniak, J.: Networking materials data: accelerating discovery at an experimental facility. In: Big Data and High Performance Computing, pp. 117–132. IOS Press (2015)
12. Foster, I., Kohr Jr., D.R., Krishnaiyer, R., Choudhary, A.: Double standards: bringing task parallelism to HPF via the message passing interface. In: ACM/IEEE Conference on Supercomputing, pp. 36–36 (1996)
13. Fryxell, B., Olson, K., Ricker, P., Timmes, F., et al.: FLASH: an adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. Astrophys. J. Suppl. Ser. **131**(1), 273 (2000)

14. Guo, H., He, W., Peterka, T., Shen, H.W., Collis, S., Helmus, J.: Finite-time Lyapunov exponents and Lagrangian coherent structures in uncertain unsteady flows. IEEE Trans. Vis. Comput. Graph. **22**(6), 1672–1682 (2016)
15. Guo, H., He, W., Seo, S., Shen, H.W., Peterka, T.: Extreme-scale stochastic particle tracing for uncertain unsteady flow analysis. Mathematics and Computer Science Division, Argonne National Laboratory (2016, preprint). http://www.mcs.anl.gov/papers/P6000-0416.pdf
16. Habib, S., Pope, A., Finkel, H., Frontiere, N., Heitmann, K., Daniel, D., Fasel, P., Morozov, V., Zagaris, G., Peterka, T., et al.: HACC: simulating sky surveys on state-of-the-art supercomputing architectures. New Astron. **42**, 49–65 (2016)
17. Herbein, S., Matheny, M., Wezowicz, M., Krogel, J., Logan, J., Kim, J., Klasky, S., Taufer, M.: Performance impact of I/O on QMCPack simulations at the petascale and beyond. In: 16th International Conference on Computational Science and Engineering, pp. 92–99. IEEE (2013)
18. Huang, H., Qin, H., Yoo, S., Yu, D.: Local anomaly descriptor: a robust unsupervised algorithm for anomaly detection based on diffusion space. In: 21st ACM International Conference on Information and Knowledge Management, pp. 405–414 (2012). http://doi.acm.org/10.1145/2396761.2396815
19. Huang, H., Qin, H., Yoo, S., Yu, D.: A new anomaly detection algorithm based on quantum mechanics. In: 12th IEEE International Conference on Data Mining, pp. 900–905 (2012). http://dx.doi.org/10.1109/ICDM.2012.127
20. Huang, H., Qin, H., Yoo, S., Yu, D.: Physics-based anomaly detection defined on manifold space. ACM Trans. Knowl. Discov. Data **9**(2), 14:1–14:39 (2014). http://doi.acm.org/10.1145/2641574
21. Iverson, J., Kamath, C., Karypis, G.: Fast and effective lossy compression algorithms for scientific datasets. In: Kaklamanis, C., Papatheodorou, T., Spirakis, P.G. (eds.) Euro-Par 2012. LNCS, vol. 7484, pp. 843–856. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32820-6_83
22. Jenkins, J., et al.: ALACRITY: analytics-driven lossless data compression for rapid in-situ indexing, storing, and querying. In: Hameurlain, A., Küng, J., Wagner, R., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) Transactions on Large-Scale Data- and Knowledge-Centered Systems X. LNCS, vol. 8220, pp. 95–114. Springer, Heidelberg (2013). doi:10.1007/978-3-642-41221-9_4
23. Koziol, Q., Podhorszki, N., Klasky, S., Liu, Q., Tian, Y., Parashar, M., Schwan, K., Wolf, M., Lakshminarasimhan, S.: ADIOS. In: High Performance Parallel I/O, pp. 203–213. Chapman and Hall/CRC (2014)
24. Ku, S., Chang, C., Adams, M., Cummings, J., Hinton, F., Keyes, D., Klasky, S., Lee, W., Lin, Z., Parker, S., et al.: Gyrokinetic particle simulation of neoclassical transport in the pedestal/scrape-off region of a Tokamak plasma. J. Phys: Conf. Ser. **46**(1), 87 (2006)
25. Kumaran, K.: Introduction to Mira. https://www.alcf.anl.gov/files/bgq-perfengr.pdf
26. Lakshminarasimhan, S., Jenkins, J., Arkatkar, I., Gong, Z., Kolla, H., et al.: ISABELA-QA: query-driven analytics with ISABELA-compressed extreme-scale scientific data. In: International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2011, pp. 1–11. ACM (2011). http://doi.acm.org/10.1145/2063384.2063425
27. Lakshminarasimhan, S., Shah, N., Ethier, S., Ku, S.H., Chang, C.S., Klasky, S., Latham, R., Ross, R., Samatova, N.F.: ISABELA for effective in situ compression of scientific data. Concurr. Comput.: Pract. Exp. **25**(4), 524–540 (2013)

28. Liu, Q., Logan, J., Tian, Y., Abbasi, H., Podhorszki, N., Choi, J.Y., Klasky, S., et al.: Hello ADIOS: the challenges and lessons of developing leadership class I/O frameworks. Concurr. Comput.: Pract. Exp. **26**(7), 1453–1473 (2014). http://dx.doi.org/10.1002/cpe.3125
29. Liu, X., Shen, H.: Association analysis for visual exploration of multivariate scientific data sets. IEEE Trans. Vis. Comput. Graph. **22**(1), 955–964 (2016). http://dx.doi.org/10.1109/TVCG.2015.2467431
30. Liu, Z., Wang, B., Wang, T., Tian, Y., Xu, C., Wang, Y., Yu, W., Cruz, C.A., Zhou, S., Clune, T., et al.: Profiling and improving I/O performance of a large-scale climate scientific application. In: 22nd IEEE International Conference on Computer Communication and Networks, pp. 1–7 (2013)
31. Malakar, P., Vishwanath, V., Munson, T., Knight, C., Hereld, M., Leyffer, S., Papka, M.E.: Optimal scheduling of in-situ analysis for large-scale scientific simulations. In: ACM International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2015 (2015)
32. Nowell, L.: Science at extreme scale: architectural challenges and opportunities (2014). http://www.mcs.anl.gov/~hereld/doecgf2014/slides/ScienceAtExtremeScale_DOECGF_Nowell_140424v2.pdf
33. Perilla, J.R., Goh, B.C., Cassidy, C.K., Liu, B., Bernardi, R.C., Rudack, T., Yu, H., Wu, Z., Schulten, K.: Molecular dynamics simulations of large macromolecular complexes. Curr. Opin. Struct. Biol. **31**, 64–74 (2015)
34. Peterka, T., Kwan, J., Pope, A., Finkel, H., Heitmann, K., Habib, S., Wang, J., Zagaris, G.: Meshing the universe: integrating analysis in cosmological simulations. In: Ultrascale Visualization Workshop, SC 2012, pp. 186–195. IEEE (2012)
35. Peterka, T., Ross, R., Nouanesengsey, B., Lee, T.Y., Shen, H.W., Kendall, W., Huang, J.: A study of parallel particle tracing for steady-state and time-varying flow fields. In: IEEE International Parallel and Distributed Processing Symposium, pp. 580–591 (2011)
36. Schendel, E.R., Jin, Y., Shah, N., Chen, J., Chang, C.S., Ku, S.H., Ethier, S., Klasky, S., Latham, R., Ross, R., Samatova, N.F.: ISOBAR preconditioner for effective and high-throughput lossless data compression. In: 28th International Conference on Data Engineering, pp. 138–149, April 2012
37. Shekhar, A., Nomura, K.I., Kalia, R.K., Nakano, A., Vashishta, P.: Nanobubble collapse on a silica surface in water: billion-atom reactive molecular dynamics simulations. Phys. Rev. Lett. **111**(18), 184503 (2013)
38. Slawinska, M., Clark, M., Wolf, M., Bode, T., Zou, H., Laguna, P., Logan, J., Kinsey, M., Klasky, S.: A Maya use case: adaptable scientific workflows with ADIOS for general relativistic astrophysics. In: ACM Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery, p. 54 (2013)
39. Tao, D., Di, S., Chen, Z., Cappello, F.: Significantly improving lossy compression for scientific data sets based on multidimensional prediction and error-controlled quantization. In: IEEE International Parallel and Distributed Processing Symposium (2017)
40. Thibodeau, P.: Coming by 2023, an exascale supercomputer in the U.S. IEEE Spectrum. http://spectrum.ieee.org/computing/hardware/when-will-we-have-an-exascale-supercomputer
41. Windstorm, P.: Fixed-rate compressed floating-point arrays. IEEE Trans. Vis. Comput. Graph. **20**(12), 2674–2683 (2014)

42. Wozniak, J.M., Armstrong, T.G., Wilde, M., Katz, D.S., Lusk, E., Foster, I.T.: Swift/T: Scalable data flow programming for distributed-memory task-parallel applications. In: 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 95–102 (2013)
43. Wu, L., Wu, K., Sim, A., Churchill, M., Choi, J.Y., Stathopoulos, A., Chang, C., Klasky, S.: Towards real-time detection and tracking of blob-filaments in fusion plasma big data. IEEE Trans. Big Data **2**(3), 262–275 (2016)
44. Zhao, Y., Wilde, M., Foster, I.: Virtual data language: a typed workflow notation for diversely structured scientific data. In: Taylor, I., Deelman, E., Gannon, D., Shields, M. (eds.) Workflows for e-Science, pp. 258–278. Springer, London (2007). doi:10.1007/978-1-84628-757-2_17